Étude numérique des écarts entre nombres premiers

Hidekela - MAFI

19/05/25

Introduction

Les nombres premiers sont les blocs de construction fondamentaux de l'arithmétique. Leur répartition soulève des questions profondes. Soit $(p_k)_{k\geq 1}$ la suite croissante des nombres premiers. On considère les écarts entre deux premiers successifs :

$$g_k = p_{k+1} - p_k,$$

et leur normalisation logarithmique :

$$r_k = \frac{g_k}{\log p_k}.$$

Un résultat asymptotique suggère que $r_k \to 1$ lorsque $k \to \infty$. L'objectif est ici d'étudier ce comportement numériquement jusqu'à 10^8 et d'explorer la loi limite des r_k.

Méthodologie numérique

Génération des nombres premiers

Le fichier `crible.py` contient deux fonctions principales :

- `cribleEratosthene(n)` : génère les nombres premiers $\leq n$ via le crible classique.
- `cribleSegmented(n)` : permet de générer efficacement les premiers jusqu'à 10^8 par blocs.

Le crible segmenté limite la consommation mémoire en divisant l'intervalle $[2, n]$ en segments de taille raisonnable :

```
def cribleSegmented(n, seg_size = pow(10, 5)):
        limit = int(sqrt(n))
        smallPrimes = cribleEratosthene(limit)

        seg_size = min(seg_size, n - limit + 1)
        a, b = limit + 1, limit + seg_size

        primes = [] + smallPrimes

```

```python
        while a <= n:
                isPrimesSegment = [1] * seg_size
                for p in smallPrimes:
                        firstMultiple = max(p * p, ((a + p - 1) // p) * p)
                        for multiple in range(firstMultiple - a, seg_size, p):
                                isPrimesSegment[multiple] = 0
                primes += [key + a for key, value in enumerate(isPrimesSegment) if value]

                a = b + 1
                b = min(b + seg_size, n)
                seg_size = b - a + 1

        return primes
```

Les nombres premiers sont ensuite sauvegardés dans un fichier `primes.data` pour traitement ultérieur.

Calcul des écarts et des ratios

Le fichier `gaps_ratios.py` traite la liste des premiers :

- `getGaps(primes)` : calcule les g_k.
- `getRatios(gaps, primes)` : calcule les $r_k = g_k / \log_{10}(p_k)$.

Les données sont sauvegardées dans `gaps.data` et `ratios.data`. Le choix du $\log_{10}$ n'affecte pas qualitativement l'analyse, car il ne change qu'à une constante multiplicative près.

Estimation de l'espérance empirique

Dans `ratiosHistogram_empiricExpectation.py`, les données sont lues puis utilisées pour :

- Afficher un histogramme des ratios.
- Calculer l'espérance empirique par `np.mean(ratios)`.

L'espérance estimée est enregistrée dans un fichier `empiricExpectation.data`. Deux expériences principales ont été réalisées :

- Pour $n = 10^7$, $\mathbb{E}_n \approx 2.3037$.
- Pour $n = 10^8$, $\mathbb{E}_n \approx 2.3028$.

Ces valeurs suggèrent-elle une convergence vers $\log 10 \approx 2.3026$?.

Résultats et visualisations

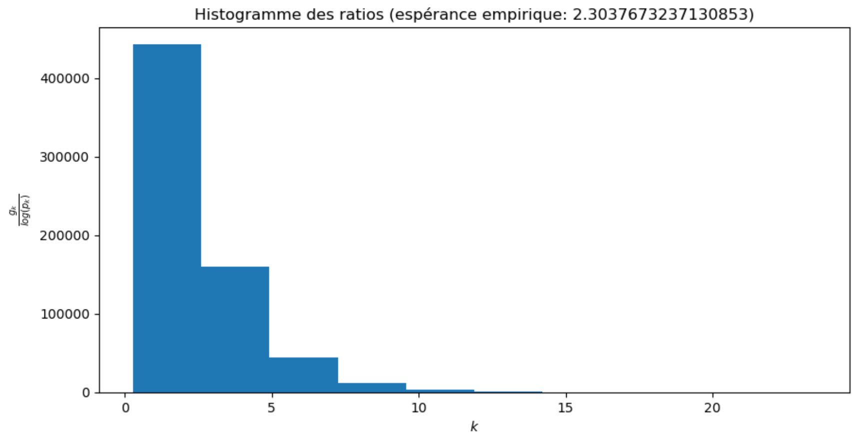

Figure 1: Histogramme pour $N = 10^7$, $\mathbb{E} \approx 2.3037$

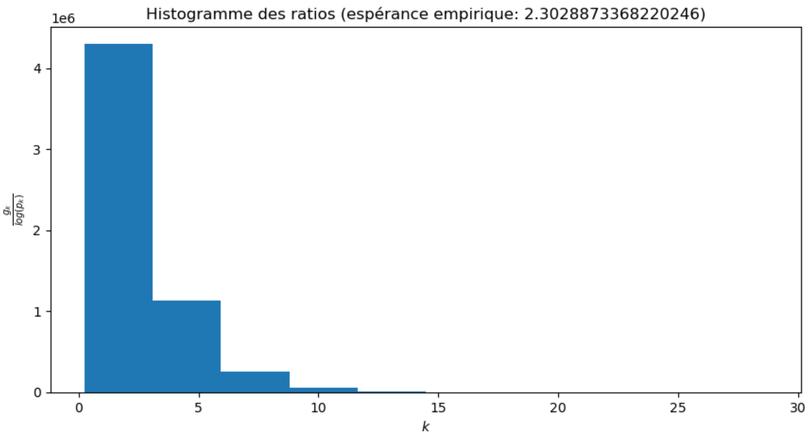

Figure 2: Histogramme pour $N = 10^8$, $\mathbb{E} \approx 2.3028$

Les histogrammes présentent une forme caractéristique rappelant celle d'une loi exponentielle.

Conjecture sur la loi limite

Le fichier `conjecture.py` compare la densité empirique des ratios à la loi exponentielle de paramètre $\lambda = 1$. La courbe théorique est tracée via `scipy.stats.expon.pdf()`.

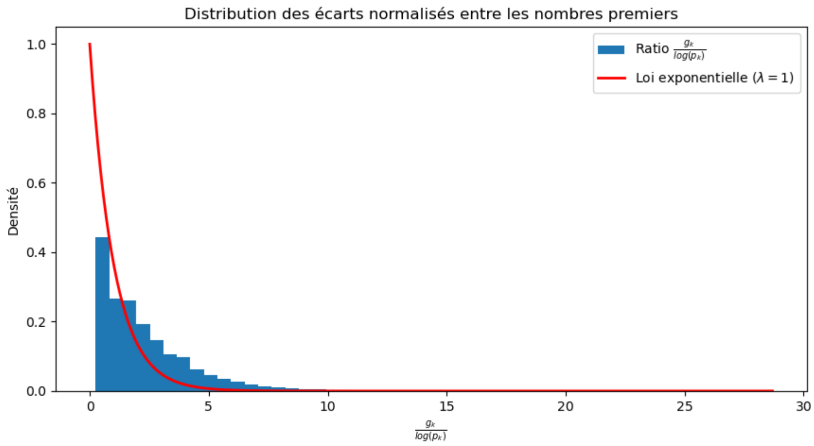

Figure 3: Comparaison avec une loi exponentielle de moyenne 1

On observe une assez bonne correspondance pour les petites valeurs, mais un léger écart pour les grandes valeurs, ce qui laisse supposer une loi « presque » exponentielle.

Discussion

Les résultats confirment que :

- Les ratios $r_k = \frac{g_k}{\log p_k}$ tendent vers une valeur moyenne $\approx \log 10$.
- Leur distribution empirique est proche d'une loi exponentielle.

Le choix du logarithme décimal (au lieu du népérien) justifie que la moyenne tourne autour de $\log 10$. Si l'on avait utilisé $\ln(p_k)$, la moyenne tendrait vers 1.

Ces résultats soutiennent l'idée que les écarts entre premiers sont distribués de façon aléatoire avec une structure probabiliste sous-jacente.

Conclusion

L'étude numérique confirme la croissance logarithmique des écarts entre nombres premiers et suggère une distribution limite exponentielle pour les ratios $g_k/\log p_k$. Les écarts semblent se comporter comme des variables aléatoires exponentielles centrées sur une moyenne constante.

Références et outils

- Langages : `Python 3`, `NumPy`, `Matplotlib`, `SciPy`.
- Données : `primes.data`, `gaps.data`, `ratios.data`, `empiricExpectation_10_7.data`, `empiricExpectation_10_8.data`.
- Figures : `conjecture.png`, `ratiosHistogram_empiricExpectation_10_7.png`, `ratiosHistogram_empiricExpectation_10_8.png`.